ESCOLA
POLITÉCNICA
DA USP

Instituto de Física da USP

SAMPA

Digital Specifications

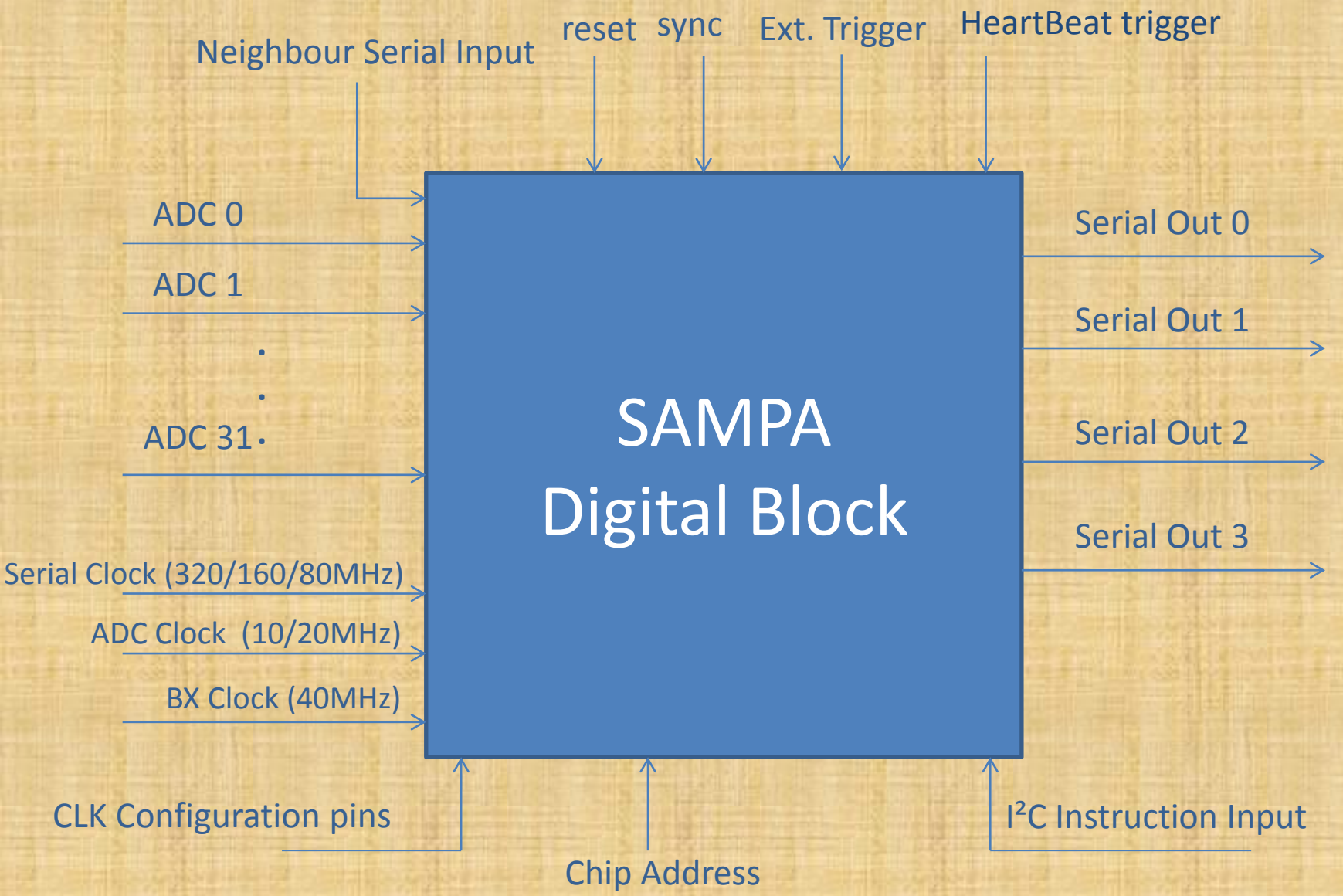
29/01/2014

Bruno Sanches
Heitor Neves

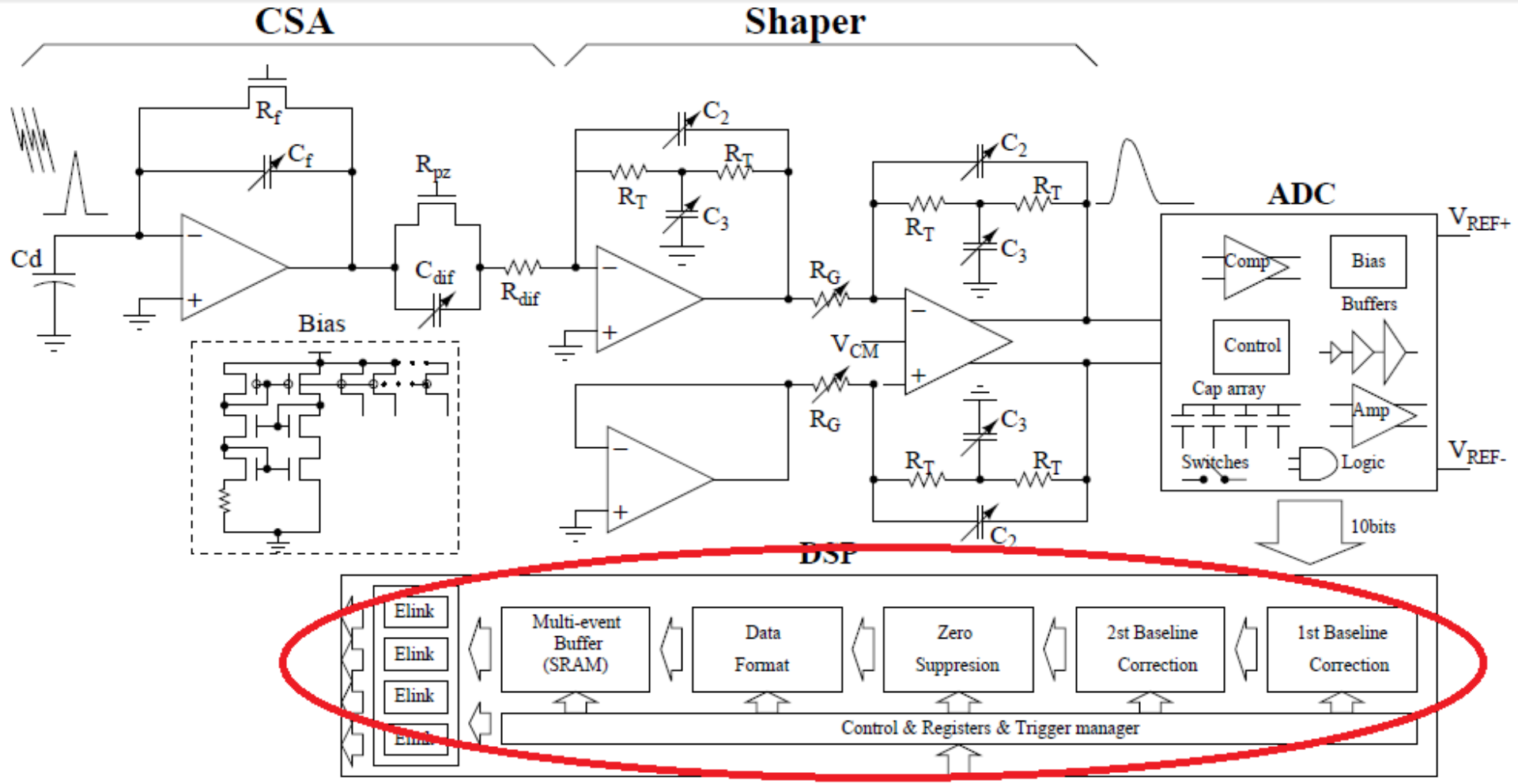
SAMPA - Characteristics

- Technology: **0.13 µm** TSMC
- Channels: **32**
- I/O: 4 serial outputs (data), 1 I^2C (inst.), 32 10-bit ADC inputs.
- Trigger: **one** external trigger or **continuous** mode
- **Daisy-chained** readout: Besides sending its own data, SAMPA is capable of sending neighbor ASIC data
- Hard wired configuration pins (chip address + clock selectors)

General Vision – SAMPA Digital Block



General Vision – SAMPA Digital Part



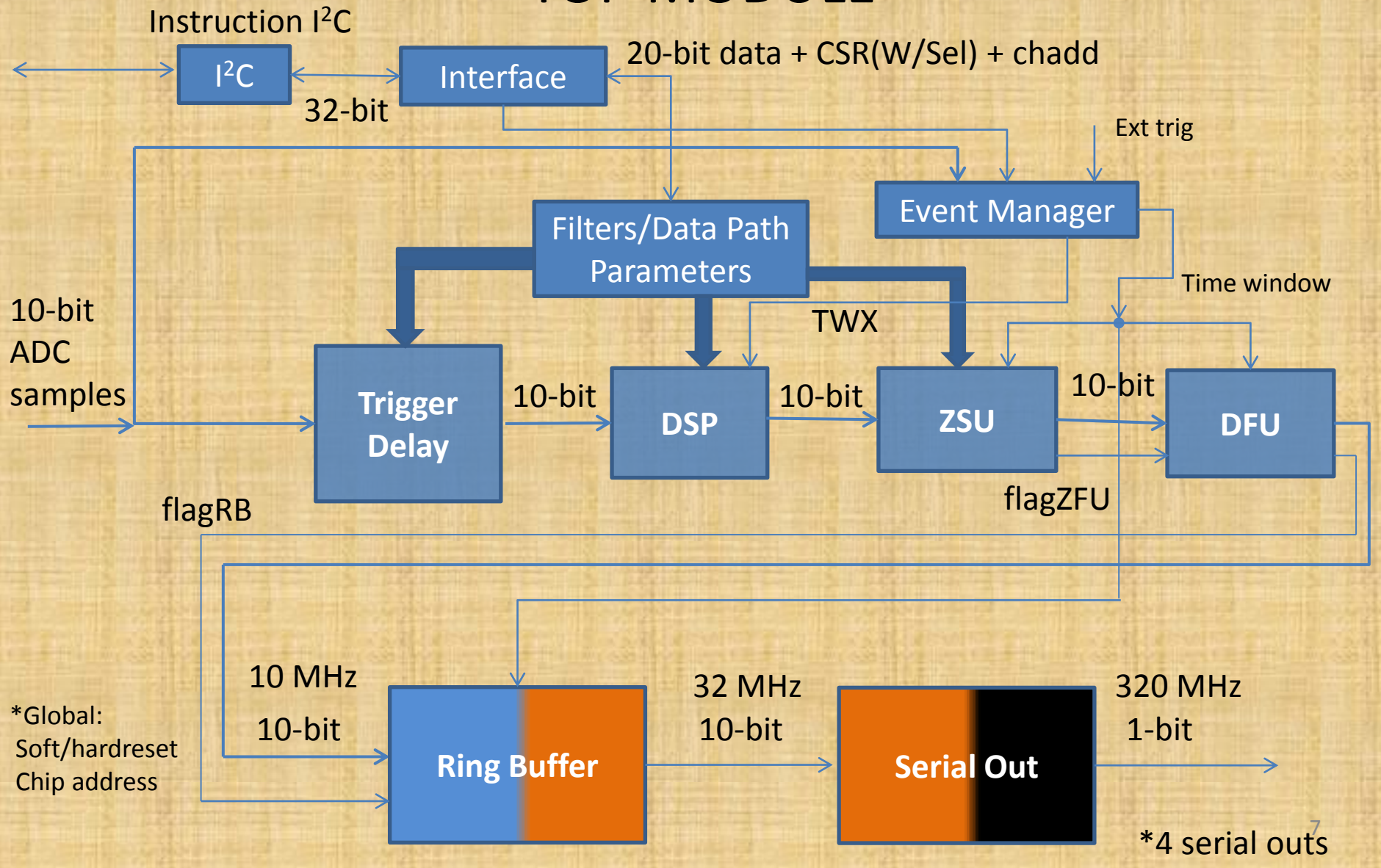
*plus tail cancelation filter

How does it work?

- **Receives** 10-bit data on 10 MHz
- **Filters the data** through a DSP (3 filters)
- **Suppresses zeros**, reducing data size
- **Formats data**
- **Stores,** up to the end of the actual event (time window)
- **Send** data automatically from:
  - Its own 32 channels
  - Neighbors (optional)

Block Diagram

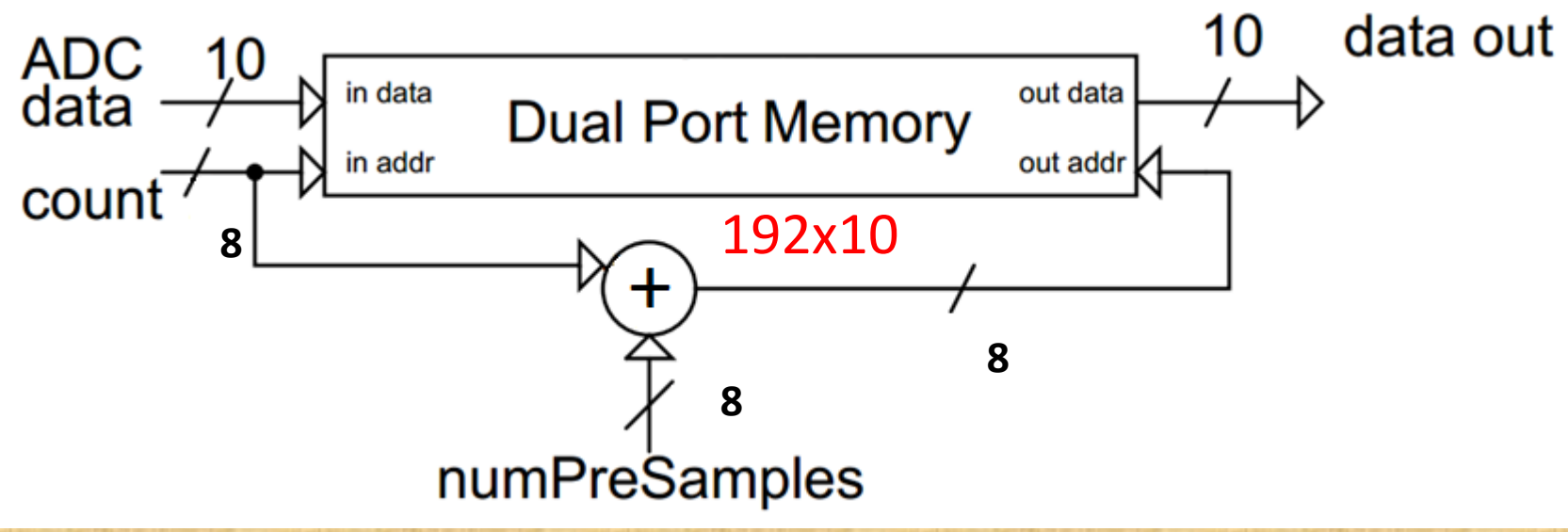
TOP MODULE



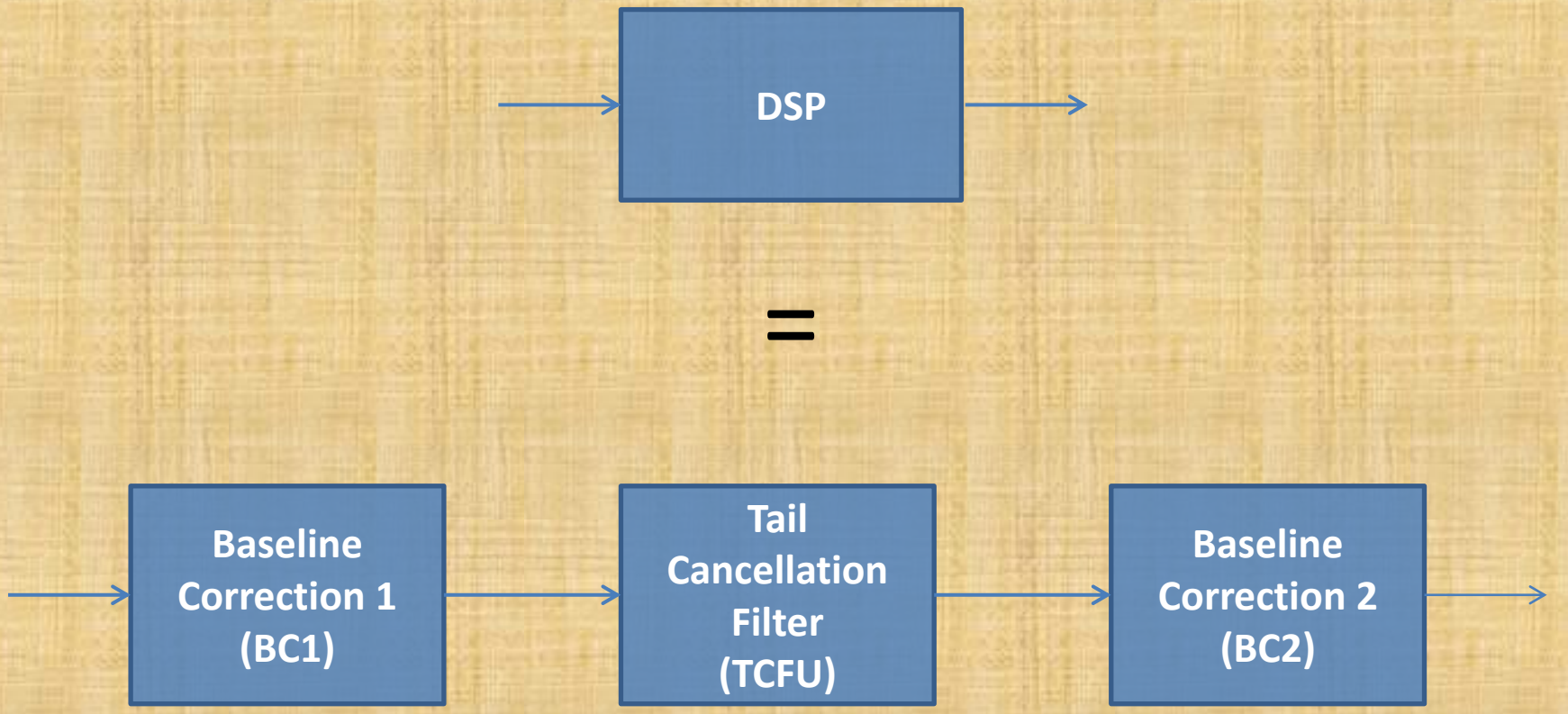
Pre-Trigger Buffer

- Inserted directly after ADC
- Delays data by a configurable amount
- Can store up to 192 samples data from before trigger
  - Satisfies 9.6us pre trigger samples of TPC for up to 20MHz sample clock

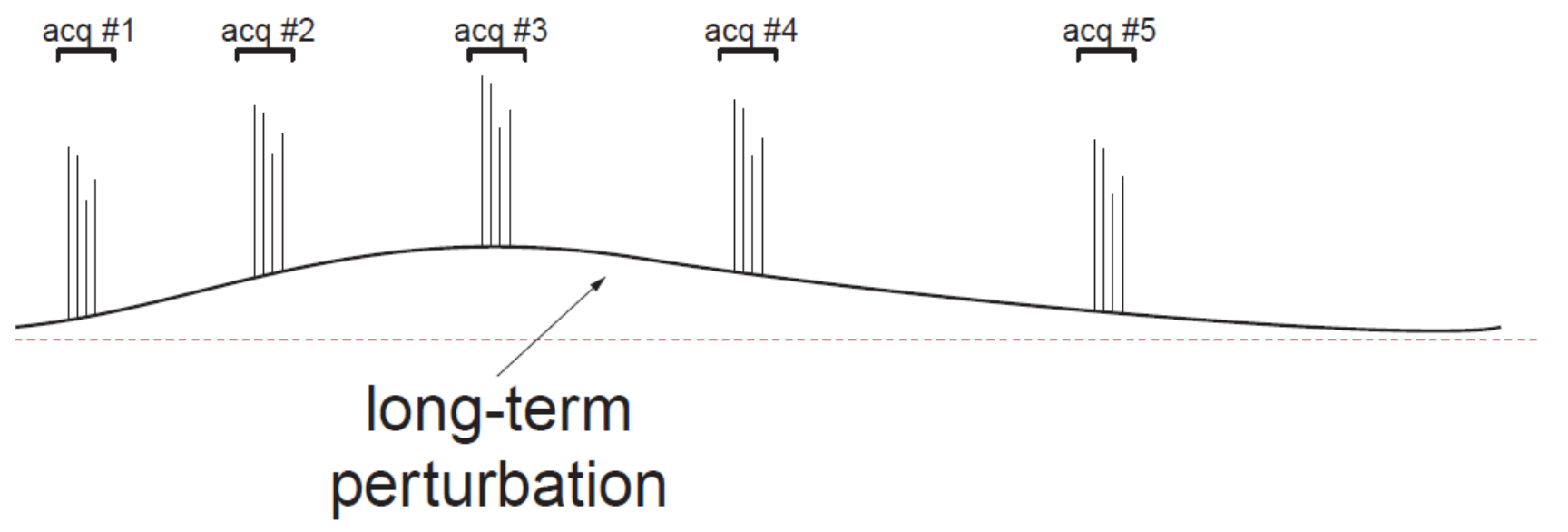
Pre-Trigger Buffer



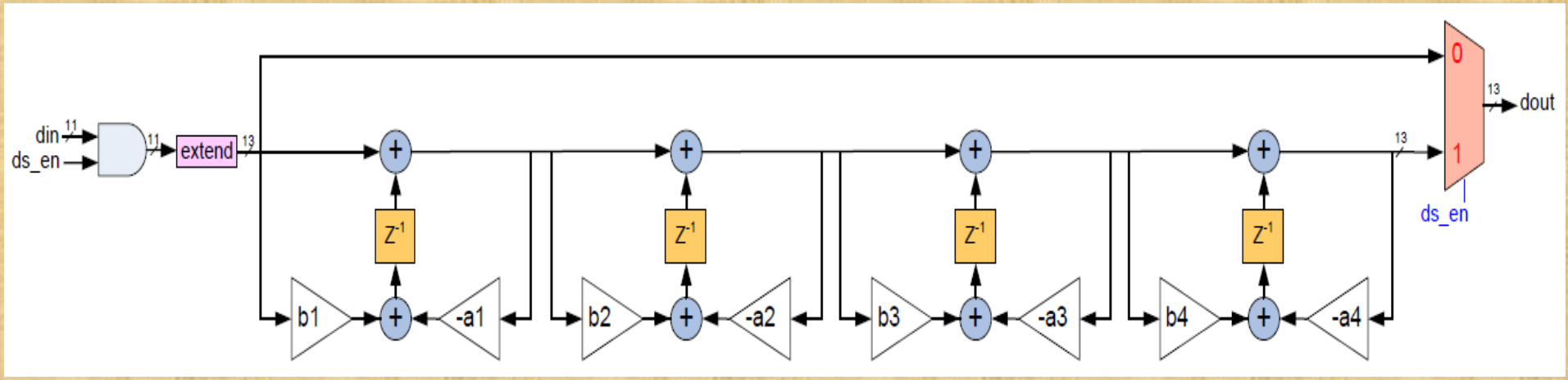
DSP


DSP
=
Baseline Correction 1 (BC1)
Tail Cancellation Filter (TCFU)
Baseline Correction 2 (BC2)

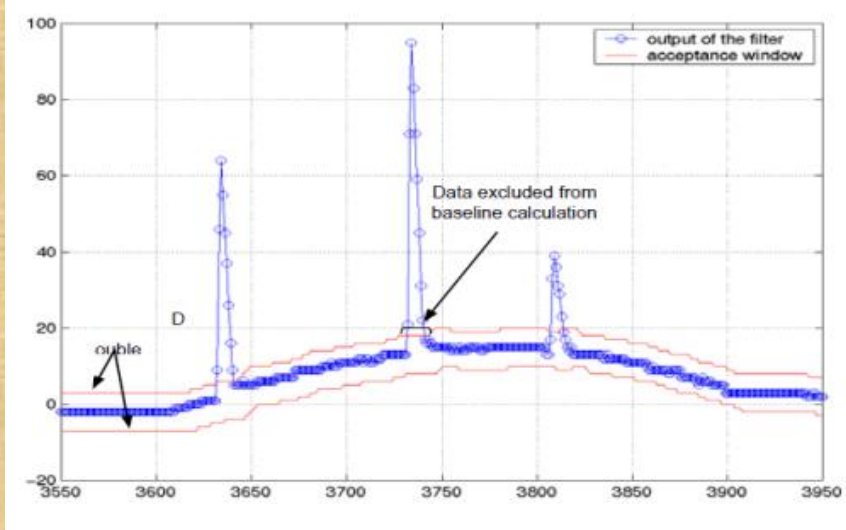
Baseline Correction 1

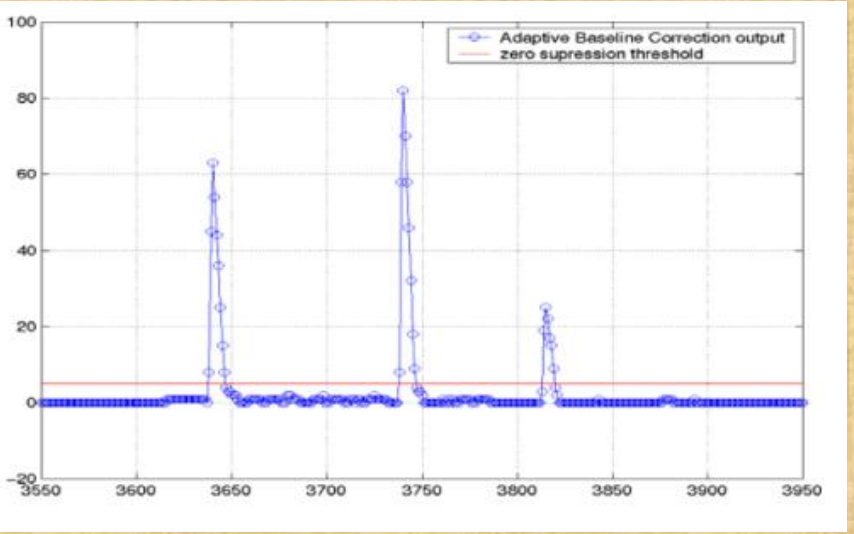


Tail Cancellation Filter

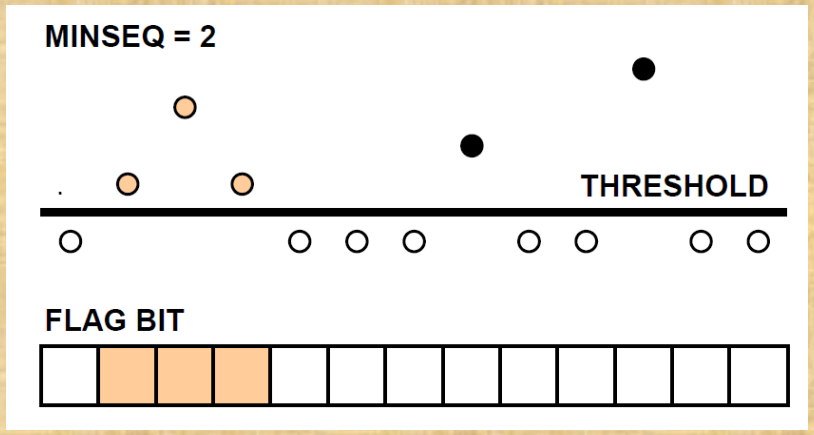


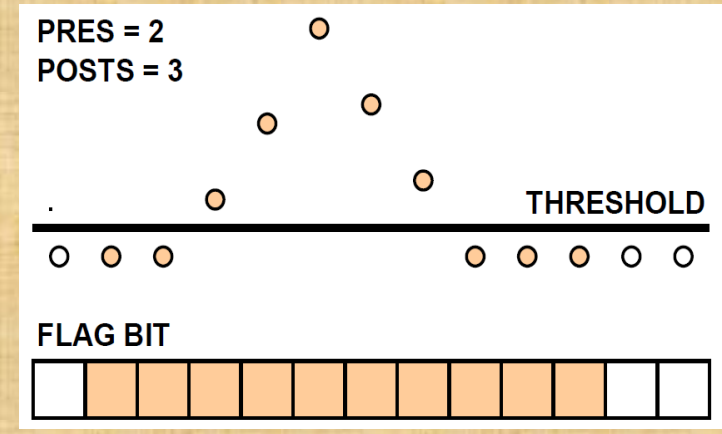
Baseline Corretion 2

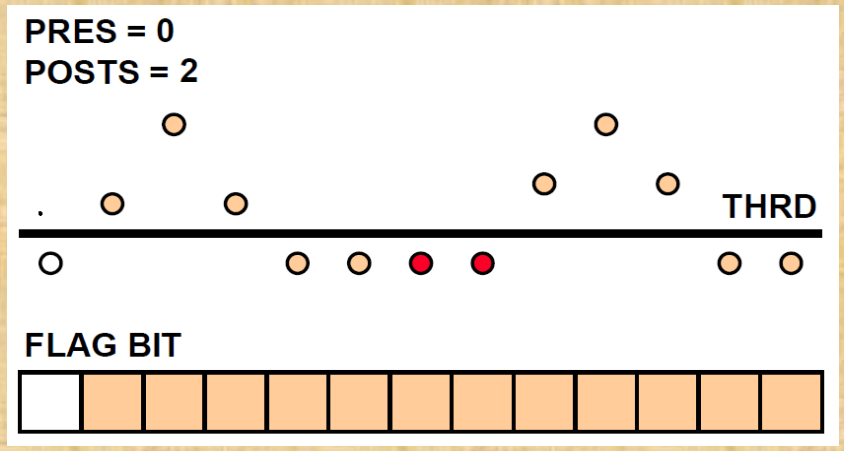




Zero Suppression







- Cluster merging => 1 or 2 samples

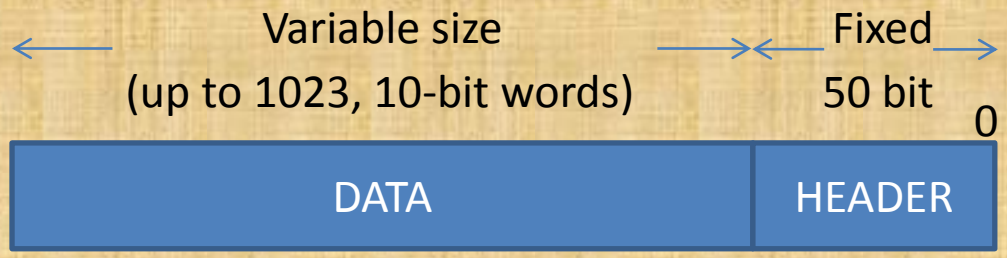
Packets Generated

- Data
  - Header + linked list data
- Neighbor
  - Header + linked list data
- Heartbeat
  - Packet indicating that the detector eletronics are operational
- Channel fill
  - No data: sends an empty packet (header only)

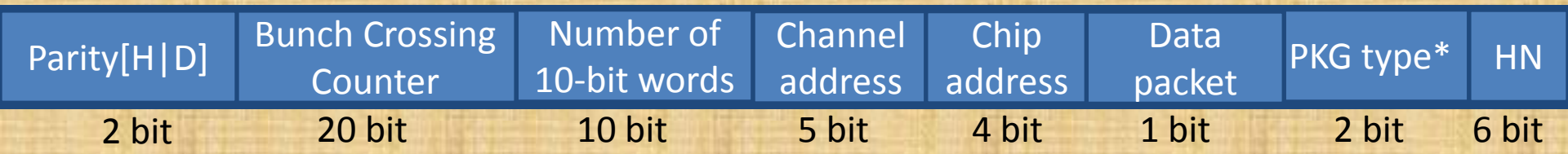
Data Format

- Data losses its temporal positioning due to Zero Suppression
- Need to add information:
  - Last samples time
  - Cluster size
- Data structure: linked list

Packet Formatting

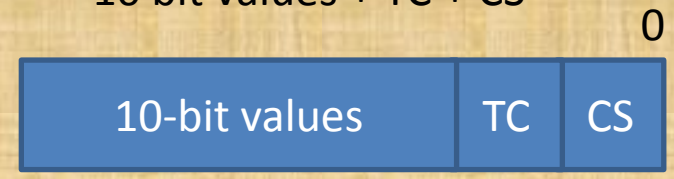


Header



Cluster

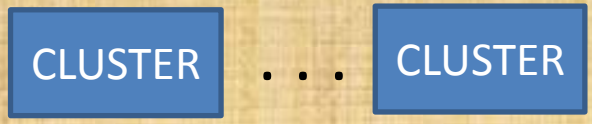
- 10 bit values + TC + CS



- CS: Cluster Size
- TC: Time Count

Data

- Data is composed of a variable amount of clusters



- **Packet type indicated by Data packet + PKG type fields ***
  - **Normal Data, Data Truncated, HeartBeat, Trigger, Sync, Channel Filler**

*Packet type codes in the next slide

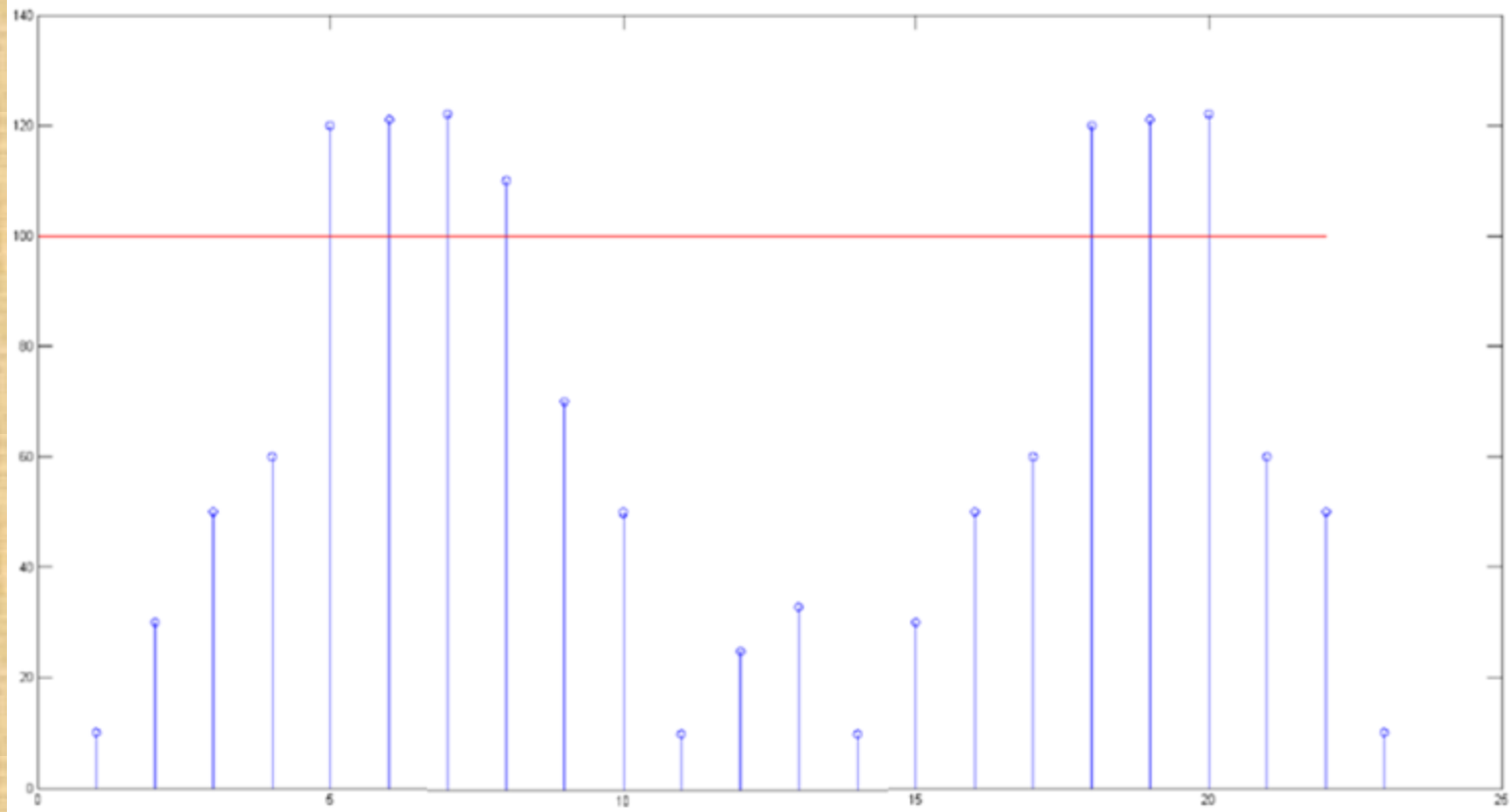
- HN are redundant bits to correct the header of the package (57 , 63)
  - 6 bits for correction and one with the parity
- ParityD is the parity of the data section of the package
- ParityH includes all bits in the header (is inserted after the encoding) -> flags if need correction!

Packet Types

Package types							Bits in header		
Data	HB	Sync	Chfill	TrigerTE	DATA Trunk	spare	Normal	PK Type 0	PK Type 1
1							1	0	0
1				1			1	0	1
1					1		1	1	0
1				1	1		1	1	1
	1						0	0	0
		1					0	0	1
			1				0	1	0
						1	0	1	1

*Spare is a unused coded packet tipe…

Data Packets Formatting - Example

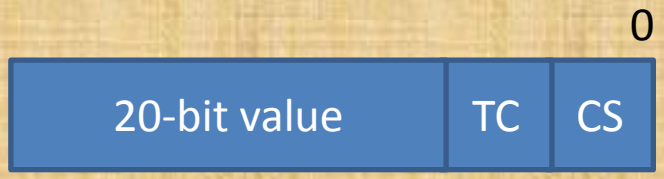


10	3	V(3)	V(4)	V(5)	V(6)	V(7)	V(8)	V(9)	V(10)	9	16	V(16)	V(17)	V(18)
V(19)	V(20)	V(21)	V(22)											

PRES = 2, POS = 2

Muon cluster sum

- Cluster size and timestamp in the same specification of normal mode
- Sum data size enough for one full time window



- CS: Original cluster Size
- TC: Time Count

Data Truncated

- Sampa chip will have 2 buffers on each channel
  - Header Memory
  - Data Memory
- The data memory will receive data until it have not free space
  - If the last packet can not be stored in the available space, it will be discarded
    - Header memory will receive a header with the data truncated option set indicating that data was lost in the buffers

Triggers

- Sync* :
  - If 0 -> 1 Resets bunch crossing counter
- HB trigger*:
  - If 0 -> 1 Generate HB packet
- External trigger*:
  - If 0 -> 1 Trigger event
    - The trigger is generated in next rising border of BX Clock

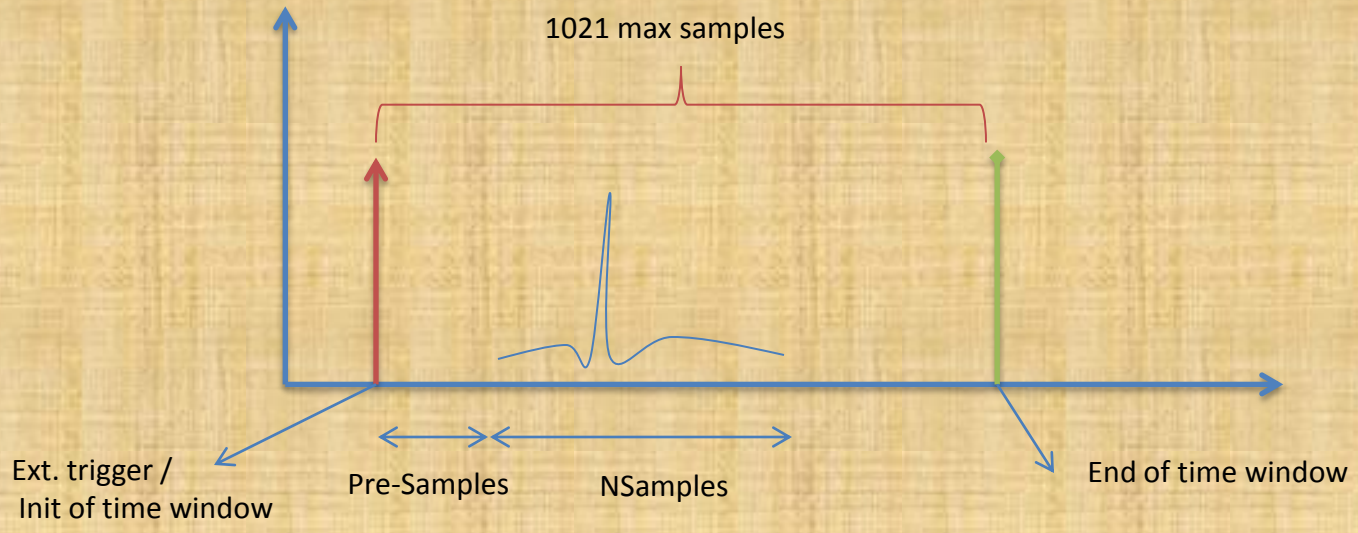
*Elink based pins synchronous with Serial clock (320/160/80)
-Signals must be valid on the falling border of serial clock
-SLVS pins!

Packets

- Sync :
  - 3 Data word packet
    - 80 bits including the header
      - [HEADER] + 1010101010 1010101010 1111001101
- Heartbeat
  - Nothing else than header
    - Number of 10-bit words = 0
- Channel filler
  - Nothing else than header
    - Number of 10-bit words = 0

Pre-samples / N samples

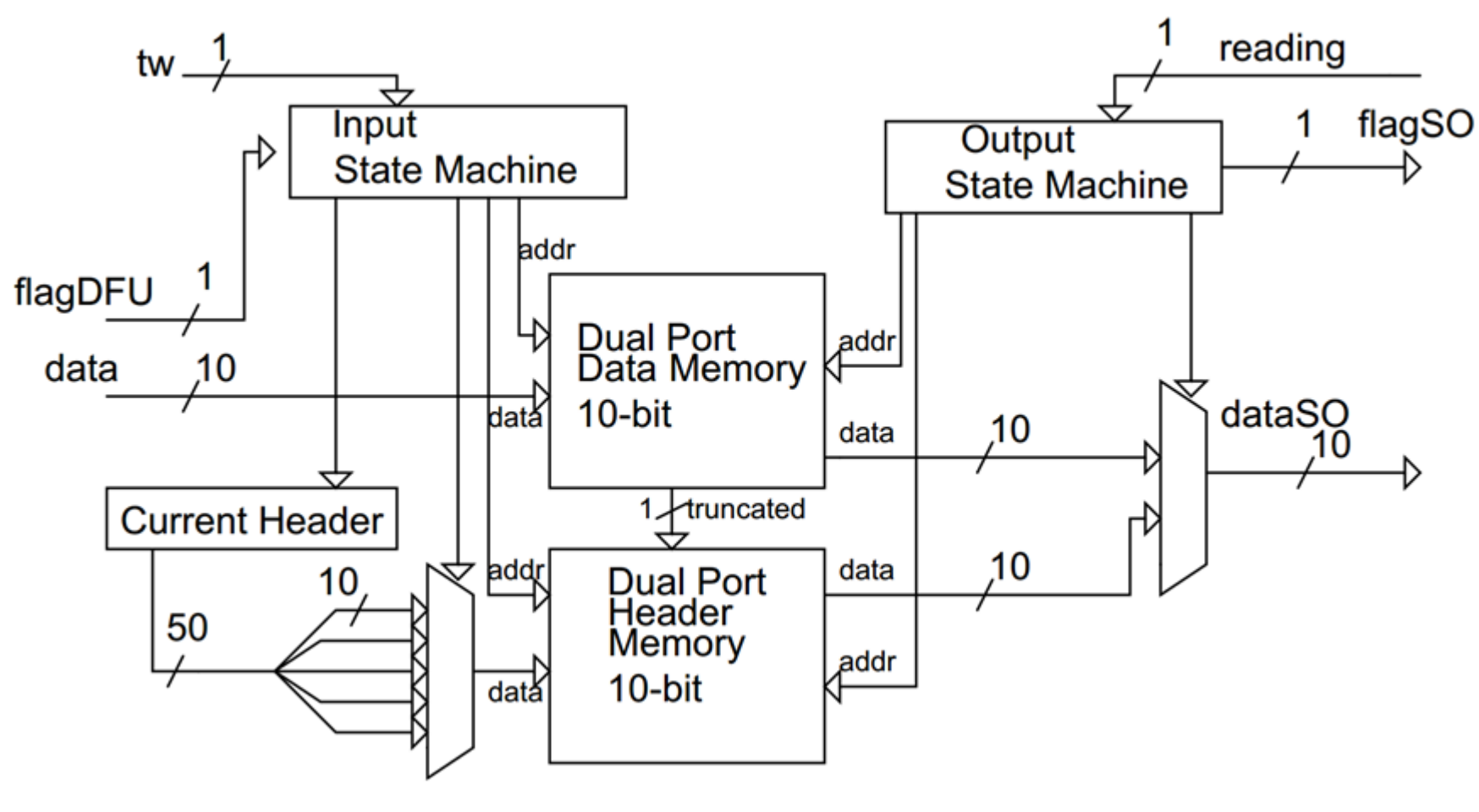
- Configurations for masking the data of the beginning / end of the time window
  - Time window still with the same interval
  - Pre-samples = number of samples to mask(0) at beginning
  - NumberOfSamples determine the interval to process before start to mask again
  - Can work in both modes (Triggered / Continuous)
  - Implemented in the Zero suppression block



Ring buffer

- Stores data from data format temporarily
- Waits for serial out to ask for the data
- Adds a header to the packet
- 2 buffers
  - Data
  - Headers
    - Headers can still be sent if data is truncated

Ring buffer


tw
1
Input
State Machine
flagDFU
1
data
10
addr
Dual Port
Data Memory
10-bit
data
addr
data
10
1
reading
Output
State Machine
1
flagSO
dataSO
10
Current Header
50
10
1
truncated
addr
Dual Port
Header
Memory
10-bit
data
addr
data
10

Serial Outs

- Checks which channels has data available
- Selects one and sends its data
- 2 clocks:
  - Serial Clock = 320 MHz (output data)
  - Serial Clock / 10 = 32 MHz (input data)
- Packets (high to low priority):
  - Heartbeat
  - Neighbor
  - Data (32, 16 or 8 channels per serial out)
  - Serial fill

Output packet sampler (Serial Out)

- The packet sources RB0 – RB31 are selected sampling the state of the buffers
  - With/without packets to send
  - The choice is time and state dependent
  - Buffer state sampling after all packets of a buffer state sample were sent
- The selection is performed alternated between the neighbor and the local channels, if there are data in both ($N_{neighbor}$: 1 relation)

Output packet sampler (Serial Out)

- Example

Initial Sample

Ch0	0
Ch1	1
Ch2	1
Ch3	0

Ch0	1
Ch1	0
Ch2	1
Ch3	1

After 1st

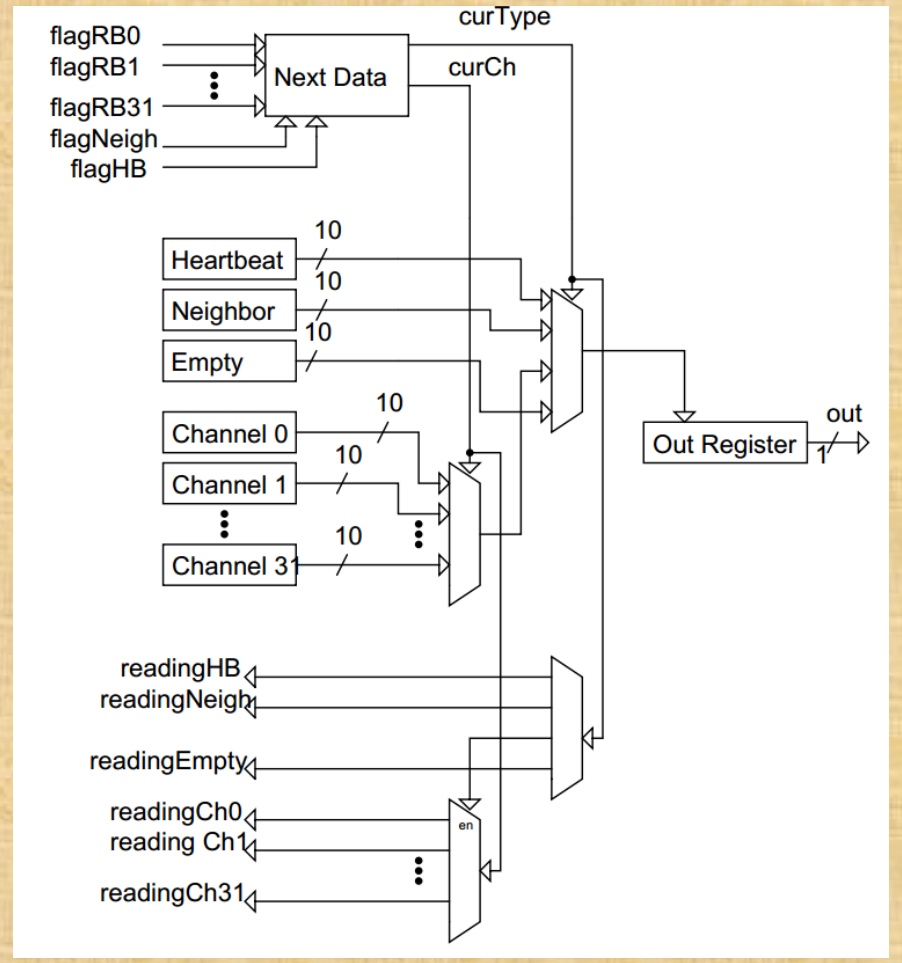
After all packets resampling

Ch0	1
Ch1	0
Ch2	1
Ch3	1

After 2nd packet

Selection should be: CH1, CH2, CH0, CH2, CH3

Serial Out


curType
flagRB0
flagRB1
flagRB31
flagNeigh
flagHB
Next Data
curCh
Heartbeat
Neighbor
Empty
Channel 0
Channel 1
Channel 31
10
Out Register
out
1
readingHB
readingNeigh
readingEmpty
readingCh0
reading Ch1
readingCh31
en

Serial Out Synchronization (BX Clock)

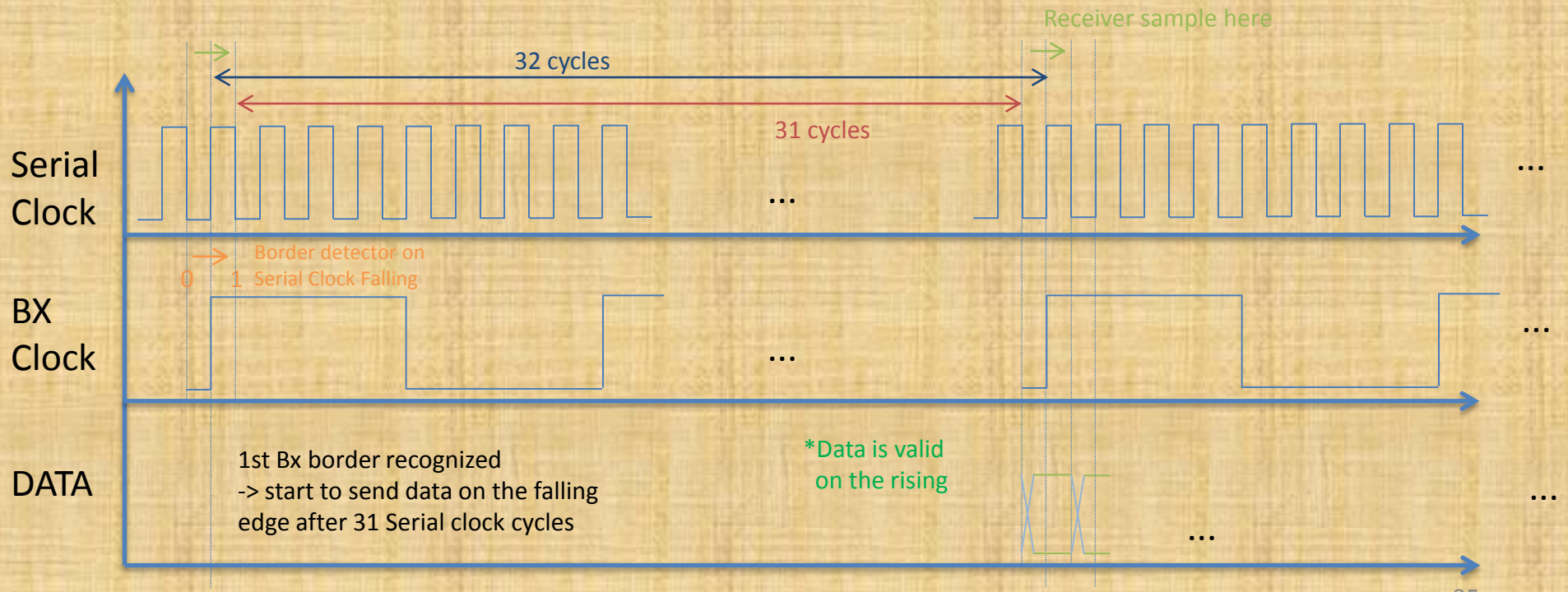
- Serial Clock and BX Clock are used to establish a valid startup bit for the serial output
- This is based on the documentation of GBTX chip
- Predictable startup delay
  - Number of clock cycles
  - After reset pin was released...

Serial Out Synchronization (BX Clock)

- SAMPA register the BX clock at Serial Clock frequency on the falling edge
- If the actual value is 1 and the last 0
  - Border detected
  - This is a rising border on Serial and BX clocks
- The data should be valid on this double rising point (this is the first bit)
  - Need some cycles to reset the medium speed part
    - Works at serial clock / 10 frequency
    - In this way: Detect the rising border of BX clock on the falling edge of serial clock
      - Wait N-1 cycles, N must be multiple of 4 and 8
      - N = 32 was selected -> wait 31 cycles (example on next slide)

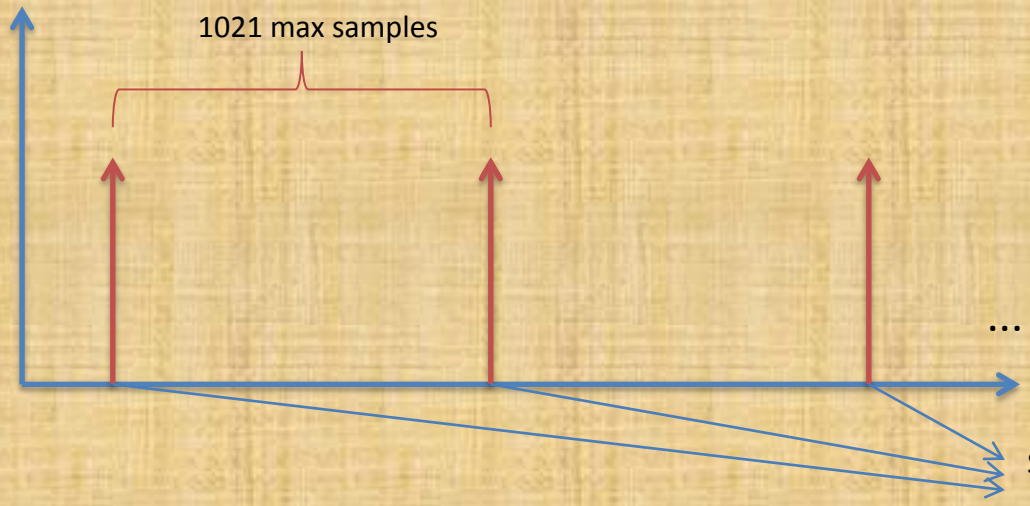
Serial Out Synchronization (BX Clock)

- The chip define the first valid bit on Serial Out evaluating both Serial Clock and GBT Clock rising/falling edges
  - This lookup happens after the first reset (start after 2 cycles)



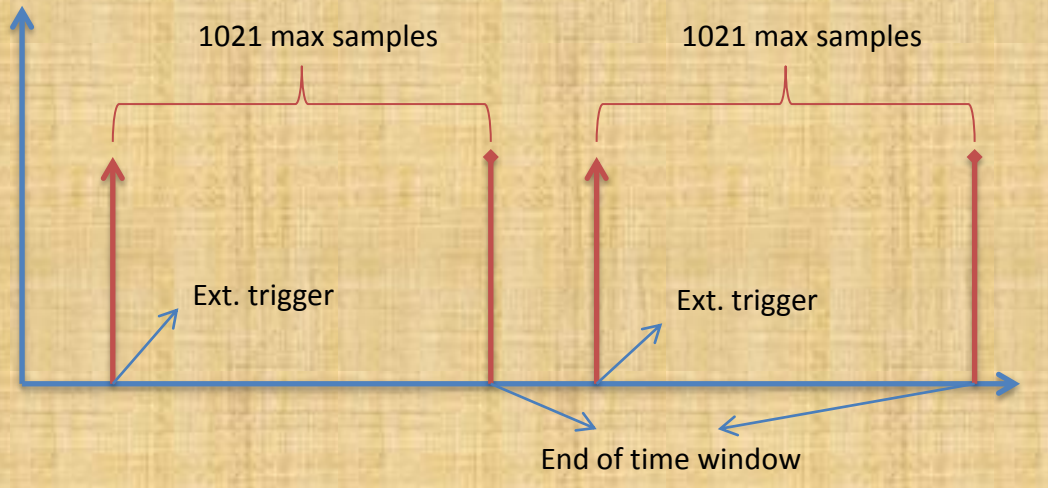
Main Operational Modes

Continuous Mode

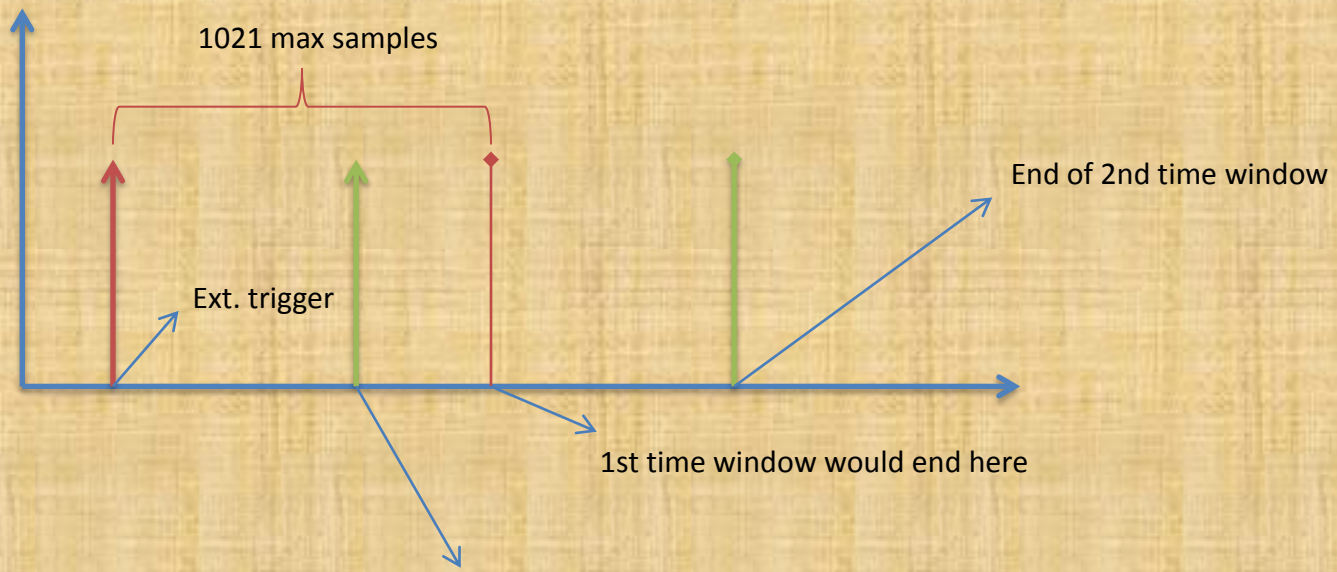


- Uninterruptable data acquisition

Triggered Mode



Triggered Mode Exception



Pedestal Runs

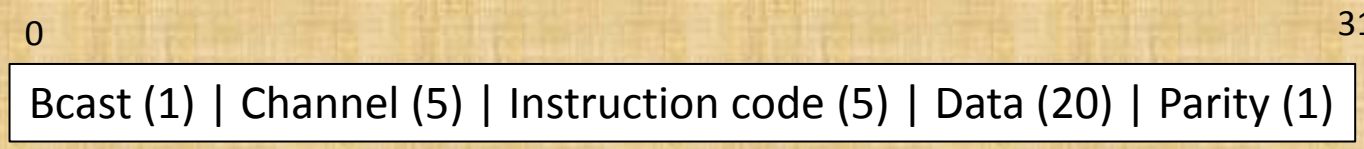
- Deactivate Zero Suppression
  - Obtaining all of the data
- Objective:
  - Determine the intrinsic detector noise
  - Calibrate filters
    - Pedestal Memory
- Can work in both modes
  - Continuous
  - Triggered

Instructions Input

- Instructions will be sent to SAMPA using an I^2C bus
  - 1MHz
  - Two dedicated normal CMOS pins
  - The I^2C address will be the same as the hardware address
    - Set hardwired by 4 pins
    - 3 other bits from the I^2C address set to a fixed pattern
    - 1111111 address reserved for broadcast
  - The I^2C address and RW bit will be used by the instruction parser

Instructions format

- The instruction will have 32 bits



  - The I^2C address and RW bit will be included in the parity calculation
  - The Bcast bit is for channel broadcast
  - The I^2C address 1111111 is for bus broadcast

Instructions Table

Instruction Name	Instruction Code [20..16] (hexa)	Register Width	Access Type	Broadcast	Description
Acessing Channel Specific Registers					
Filter Coeficient K1	00	13	RW	YES	Read or write the filter coeficient K1
Filter Coeficient K2	01	13	RW	YES	Read or write the filter coeficient K2
Filter Coeficient K3	02	13	RW	YES	Read or write the filter coeficient K3
Filter Coeficient K4	03	13	RW	YES	Read or write the filter coeficient K4
Filter Coeficient L1	04	13	RW	YES	Read or write the filter coeficient L1
Filter Coeficient L2	05	13	RW	YES	Read or write the filter coeficient L2
Filter Coeficient L3	06	13	RW	YES	Read or write the filter coeficient L3
Filter Coeficient L4	07	13	RW	YES	Read or write the filter coeficient L4
Zero Suppression Offset and Threshold	08	20	RW	YES	Read or Write ZS offset and threshold
Zero Suppression Configuration	09	07	RW	YES	Read or write ZS configuration
BC1 Subtraction Pedestal	0A	20	R(VPD) +RW	YES	Read VPD and read/write FPD
Channel Noise	0B	10	RW	YES	Read or write channel noise
Pedestal Memory Data	0C	20	RW	YES	Read or write PM (not a register, data path only)
Acessing Global Registers					
Pedestal Memory Address	0D	12	RW	N/A	Read or write PM address commom to all registers
BC2 High and Low Threshold (MAF)	0E	20	RW	N/A	Read or write upper and lower threshold of the MAF
Pre-Samples and Samples/Event	0F	17	RW	N/A	Read or write number of pre-samples and samples per event
Data Path Configuration	10	12	RW	N/A	Read or write MAU + BSU configuration
DS Enable + Power Saving + Continuous Mode + Pedestal Mode + #eLinks	11	15	RW	N/A	Continuous threshold, enable continuous mode, Enable digital shaper, power saving , pedestal on = 1, Number of active eLinks (0 = 1 eLink, 1 = 4 eLinks).
BC1 High and Low Threshold and "n"	12	14	RW	N/A	Read or write upper,lower threshold and "n" of BC1 N???
Trigger Count	13	16	R	N/A	Read trigger count
Chip Address	14	08	R	N/A	Read chip address
Error bits	15	02	R	N/A	Read error bits
Bunch Crossing Counter	16	20	R	N/A	Read the bunch crossing counter value
Commands		**Instruction Width***			
Software Trigger	1B	28	N/A	N/A	Triggers the readout
Clear Trigger Counter	1C	20	N/A	N/A	Set trigger count to 0
Clear Error Bits	1D	20	N/A	N/A	Clears error sticky bits
Reset Bunch Crossing Counter	1E	20	N/A	N/A	Resets the bunch crossing counter to 0
Software reset	1F	20	N/A	N/A	Resets whole chip, but does not erase configuration registers

*Plus: test input step (en/go), set dac value

*Channel disable, MAF config

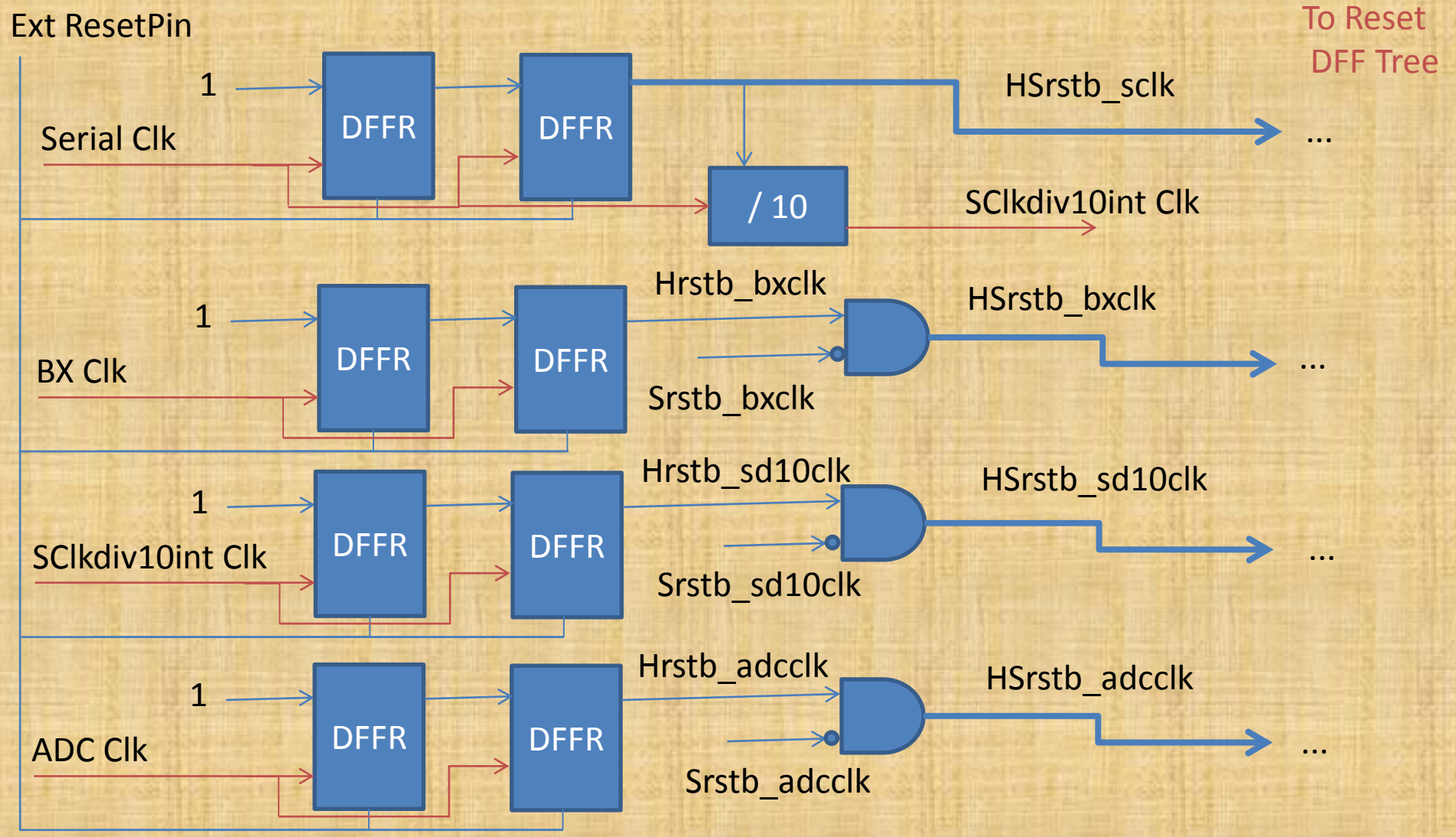
Channel Configuration

- Per channel:
  - Can be turned off by instruction (mask input and Rbflag, reset RB pointers, set power save)
  - Zero suppression threshold
  - Constant pedestal value
  - Full pedestal memory
  - Filter coefficients
- Per Chip
  - Triggered / Continuous mode
  - BC2 High and Low Threshold (MAF)
  - Pre-Samples and Samples/Event

RESET

- Hardware reset
  - Resets everything, including the clock manager
  - Asynchronous reset pin connected to Hardware reset net
    - Active low
    - The reset will be automatically synchronous deasserted
    - One reset synchronization circuit for each clock
      - Serial Clock, Serial Clock/10,BX Clock and ADC clock
- Software reset instruction
    - Reset everything (except registers and pedestal memory)
    - Send sync package
- Power on Reset
  - Automatic reset the device when powered (Hardware Reset)

HW Reset Schematic

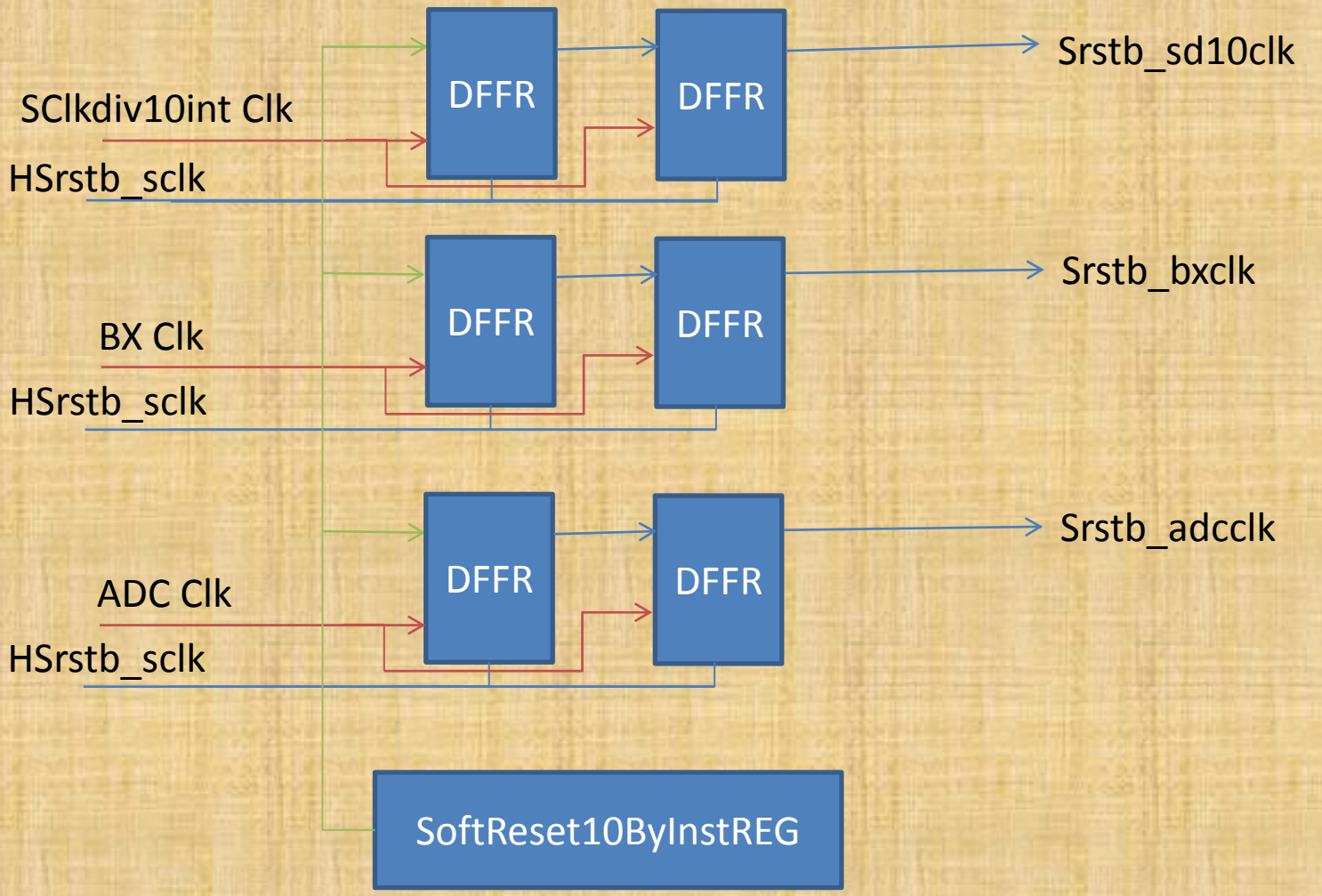


•Software Triggers Srstb_ described on next slide

*DFFR is active low and assync. reset

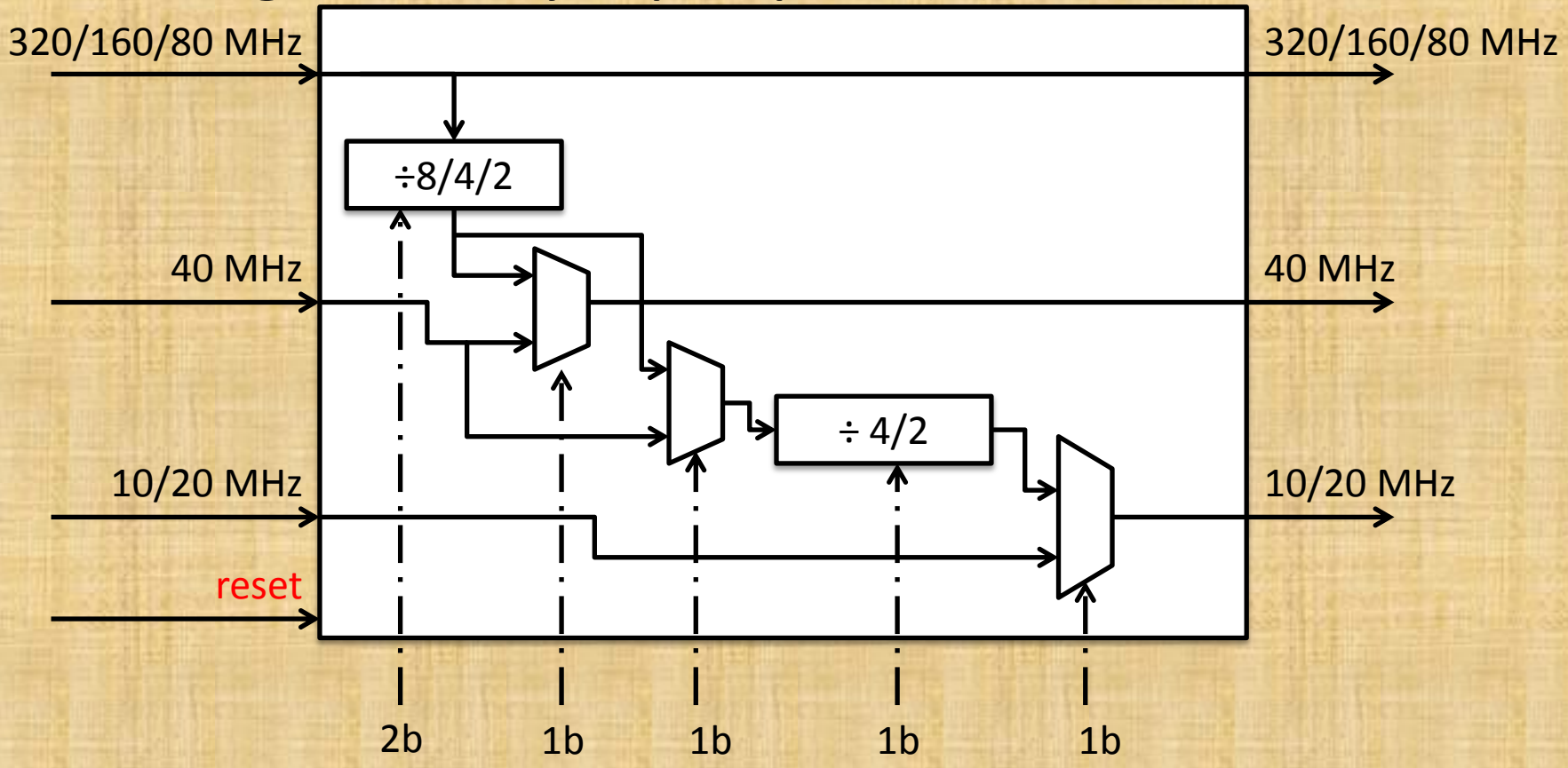
*SClkdiv10int is an internal divisor output

Soft Reset Schematic – Activated By INstruction



Clock Manager – RESET

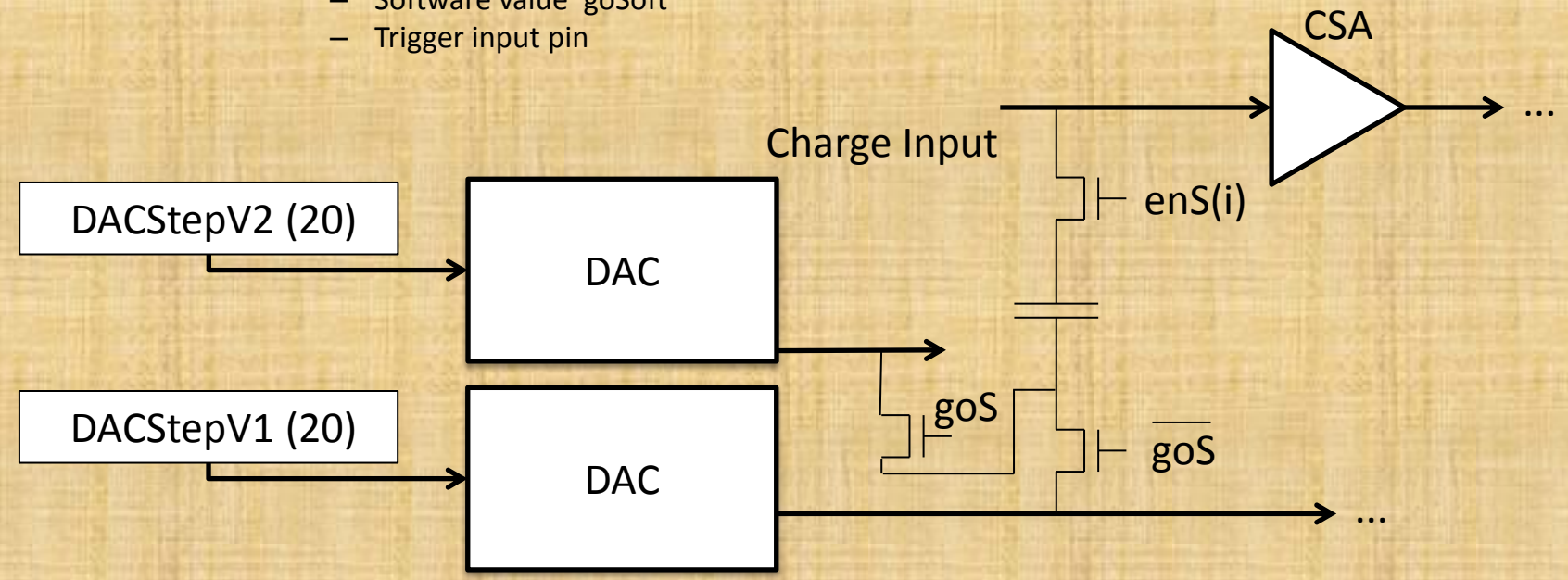
- Configurable by input pins (6)



* Reset is activated with the internal reset signal of the blocks running at Serial Out Clock frequency

Step Input Test

- Module to internaly perform a configurable step function on the amplifier input
  - 32 enable channels – enS(i)
  - 1 go signal (common to all channels) - goS
  - 2 DAC config 20 bit register
  - 1 instruction to enable a goS masking
    - goS = goSoft OR (goMask AND TRIGGERINPUTPIN)
      - Software value goSoft
      - Trigger input pin

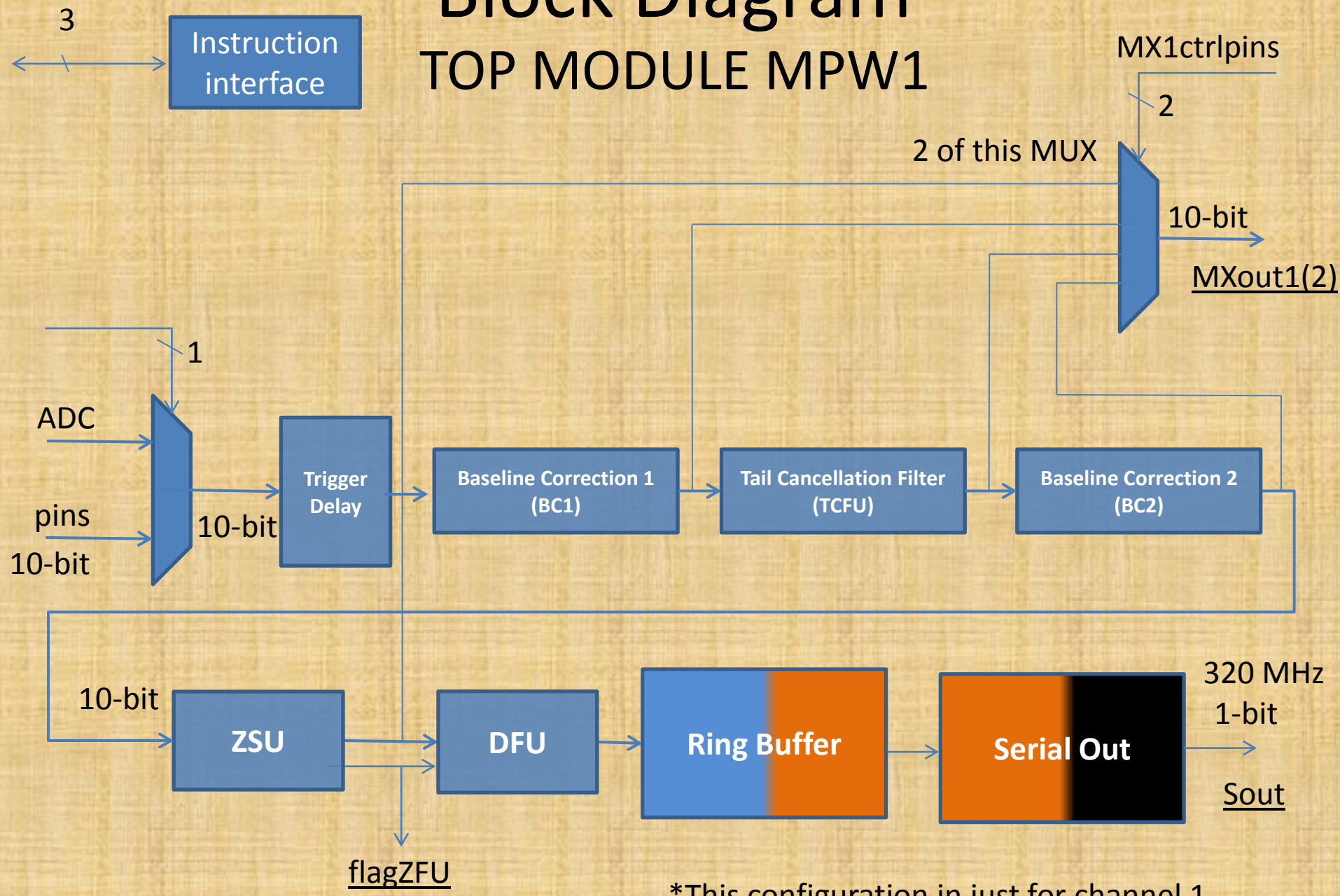


MPW 1

- Send the digital part of SAMPA including the serial outputs and buffers
  - Just 3 channels of DSP + ZSU
    - Insert memory only for active channels
- Provide a input selector
  - From ADC or external input
- Provide output selectors
  - Two 10 bit outputs + ZFUflags
    - Parallel 10 bit output for each filter + ZFU
    - External control bits for the output probe selectors

Block Diagram

TOP MODULE MPW1



*This configuration in just for channel 1

MPW1

- MPW1 will not generate heart beats
- No neighbor buffer / input
- Clocks from input pins (no internal clock manager)
- No software reset

- MPW1 will have external inputs to select the active number of serial out channels

- Will be inserted (as many as possible) 10k shift register based on the smallest flip flop of the technology

- 1 bit to bypass each filter

- 1 bit to bypass zero suppression

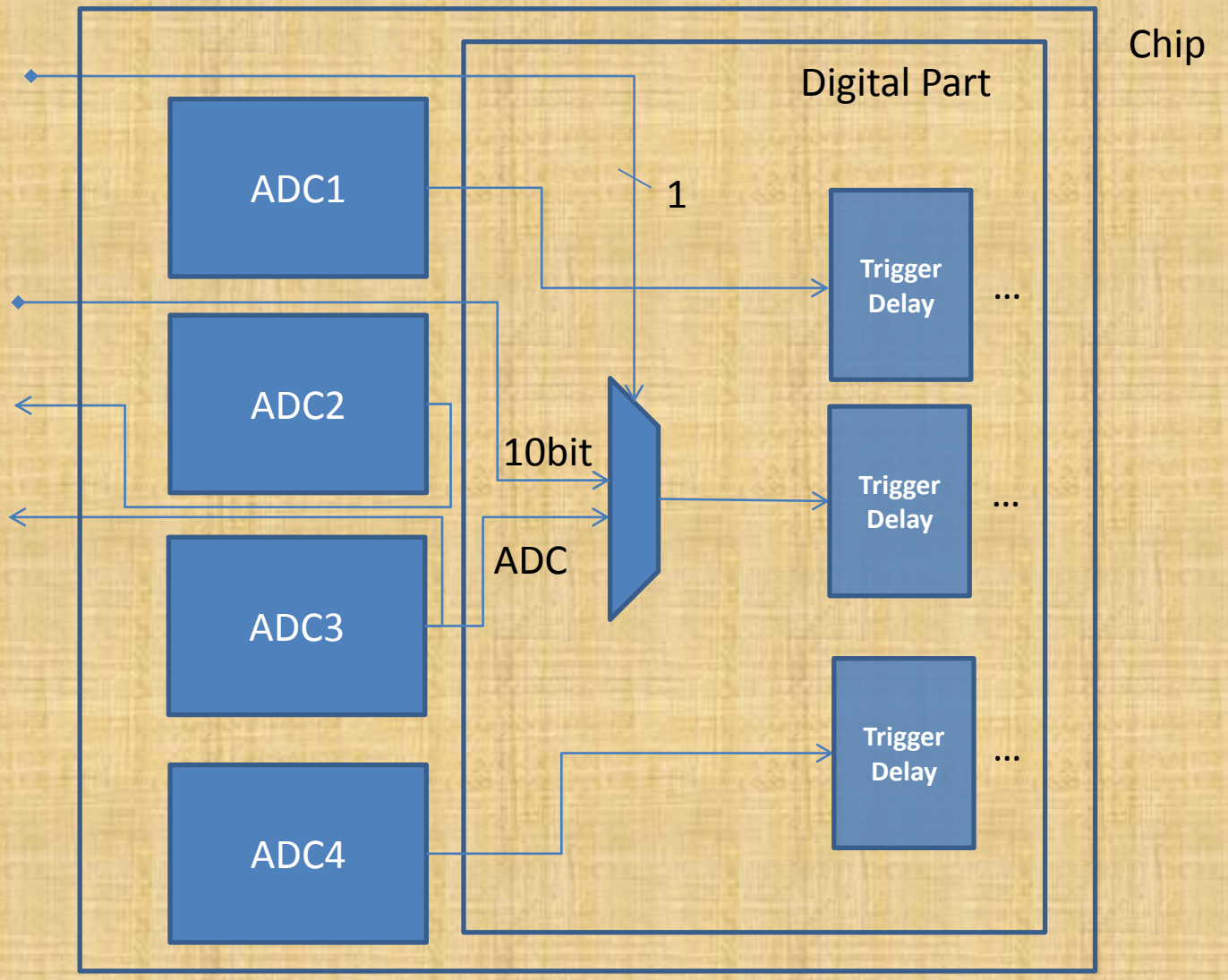
- 2 External bits to configure the number of serial outs

- 2k word memory on Ring buffer

MPW1 Reset Config

- The startup configuration of SAMPA will be
  - Triggered mode
  - Zero suppressed (low level)
  - Just serial out 0
  - Default configuration for the filters…
  - Disable tail cancelation
  - Remove serial clock sync (startup 40MHz scan)

Main Connections with analog part



TODO*

- From scratch
  - I^2C
  - Step test on input based on DAC
  - Adjusts to measurements in MPW1
    - Test points and Muxes
    - Input select

- Need Changes
  - RB
  - Serial OUT
  - Clock manager
  - Interface (Instruction Parser)
    - Change/add instructions
    - Adjust to I2C compatibility


Thank you!
keep cool